NRL/FR/5535--93-9589

The Damage Control Information Display System for the Ex-USS *Shadwell*

DAVID L. TATE

Human Computer Interaction Laboratory Branch
Information Technology Division

September 30, 1993

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 30, 1993	Final 1 Oct. 92-30 Sept. 93

4. TITLE AND SUBTITLE

The Damage Control Information Display System for the Ex-USS *Shadwell*

5. FUNDING NUMBERS

PE - 62121N
PR - RH21S22
TA - 5

6. AUTHOR(S)

David L. Tate

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/5535--93-9589

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

The Damage Control Information Display System (DCIDS) is a graphical information-retrieval and equipment-control system that gives shipboard damage control personnel the ability to detect, analyze, and combat various types of damage control situations. A prototype of DCIDS has been developed for use aboard the ex-USS *Shadwell*, the Navy's full-scale fire research and test ship from which damage control and ship survivability investigations, analyses, and evaluations are performed. This DCIDS implementation runs on a PC-compatible computer under MicroSoft Windows and communicates with the *Shadwell*'s data collection computer through an Ethernet TCP/IP connection. This report describes the operation and use of the DCIDS program. It also provides the user interface and information management specifications for modifying the program for other fire test scenarios and equipment interfacing.

14. SUBJECT TERMS

Damage Control Information Display System (DCIDS)
Damage control
Graphical user interface
Computerized control systems
Human-computer interaction
Ex-USS *Shadwell*

15. NUMBER OF PAGES

25

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

CONTENTS

THE DAMAGE CONTROL INFORMATION DISPLAY SYSTEM FOR THE EX-USS *SHADWELL*

INTRODUCTION

To conduct shipboard damage control (DC) operations effectively, a ship's crew must have the ability to detect, display, and analyze sensor information quickly and succinctly. Current damage control research includes the development of advanced sensors that will provide very detailed information from sensors for smoke, heat, flame, fire classification, flooding, hull damage, and chemical, biological, and radiological (CBR) agents [1]. The Damage Control Information Display System (DCIDS) is a graphical information-retrieval and equipment-control system that gives the operator the ability to effectively manage these types of DC resources through the use of an intuitive, interactive graphical user interface [2].

The ex-USS *Shadwell* is the U.S. Navy's full-scale fire research and test ship from which damage control and ship survivability investigations, analyses, and evaluations are performed. The *Shadwell* is highly instrumented and controlled for acquisition, processing, and storage of fire test data. Up to 400 sensors can measure temperature, obscuration, heat, radiation, gaseous products, air and liquid velocity and pressure, and hatch closure state can be connected to the ship's data acquisition computer system. Because of its ability to host full-scale tests and to provide detailed sensor data, the *Shadwell* serves as both a significant training tool and an important research tool.

The integration of the DCIDS concept and the *Shadwell*'s myriad of sensors provides an opportunity to test and evaluate the effectiveness of DCIDS as an operational concept, and it provides researchers with a tool that can help them better understand the complex interactions and dynamics of fire propagation.

BACKGROUND

The DCIDS for the *Shadwell* was developed through a cooperative Navy effort as a means of obtaining the common goal of collecting, monitoring, and controlling large amounts of information from various types of damage control sensors. The *Shadwell* was modified to support the distributed architecture proposed by DCIDS, and a specific implementation of DCIDS was created to meet the requirements of the *Shadwell*'s configuration

Ex-USS *Shadwell* Configuration

The *Shadwell* uses a MassComp computer as its primary data acquisition and processing facility. The MassComp has the capacity for up to 400 sensors of various types that can be positioned in arbitrary locations throughout the ship during fire tests. The primary method of observing sensor

Manuscript approved September 22, 1993.

outputs during a test is to configure the MassComp's data acquisition program to display the data in strip-chart form on the graphics terminals in the *Shadwell*'s control room. Sensors are connected to the input/output (I/O) channels on the MassComp, and I/O channel numbers are assigned to the various graphics displays. Because multiple channels are plotted on each display, it is sometimes difficult to get an accurate real-time assessment of the data. These data are also stored on disk, allowing non-real-time retrieval after completion of the test.

To accommodate a growing diversity of computer systems and variety of data requirements by the users, an Ethernet local area network (LAN) was installed on the *Shadwell* [3]. Ethernet was chosen because it is widely supported and available for UNIX, MS-DOS, Macintosh, and VAX systems. Standard networking functions such as rapid data transfer, file sharing, and remote login capabilities are supported. The Ethernet uses RG-58 coaxial cable, commonly referred to as ThinNet, as its transmission medium. Users can connect their computer to the *Shadwell*'s network through a standard ThinNet interface after being assigned an Internet Protocol (IP) network address by the ship's network administrator.

The installation of Ethernet on the *Shadwell* provides a distributed computing environment in which numerous systems with diverse functions and capabilities can be connected together. Figure 1 shows the Ethernet configuration planned for the *Shadwell* and the various damage control systems that can take advantage of this capability.

The *Shadwell*'s MassComp computer performs the sensor monitoring and data recording functions during full-scale fire tests. Historically, these data were available in real time only through special-purpose shared memory applications running on the MassComp concurrently with the data collection program. With the addition of the EtherNet LAN, a new method of real-time data extraction is possible [4]. A separate background process has been written for the MassComp that supports real-time sensor monitoring and data transfers across the EtherNet. This new approach to providing real-time data access allows users to create their own special-purpose application on any computer platform with any operating system that supports EtherNet networking.

Damage Control Information Display System

The Damage Control Information Display System was developed to provide advanced techniques in information management and control for current and developmental damage control sensor systems. To support the development of advanced sensor technology and provide the technology "push" to ease the integration of these technologies into operational systems, a graphical user interface design was developed to improve the performance of DC personnel through the use of intuitive, easy-to-use graphical displays and controls [2]. The DCIDS concept uses high-resolution large-screen color monitors to display information in multiple graphical formats to give operators a flexible and easy means of viewing sensor information. Special techniques are used to present data with visual cues that are rapidly recognized, thus allowing complex data to be more quickly assimilated. Direct manipulation techniques provide an intuitive user interface, assuring the operators that they control the system, rather than serve it.

In addition to providing an intuitive graphical user interface to DC personnel, the DCIDS architecture includes the use of a distributed information-processing architecture for sensor and console interconnections. In this configuration, multiple DC consoles are connected to multiple sensors via interface units on a local area network, as shown in Fig. 2. This provides flexibility and reconfigurability for both the functional duties of DC personnel and the physical location of DC equipment.

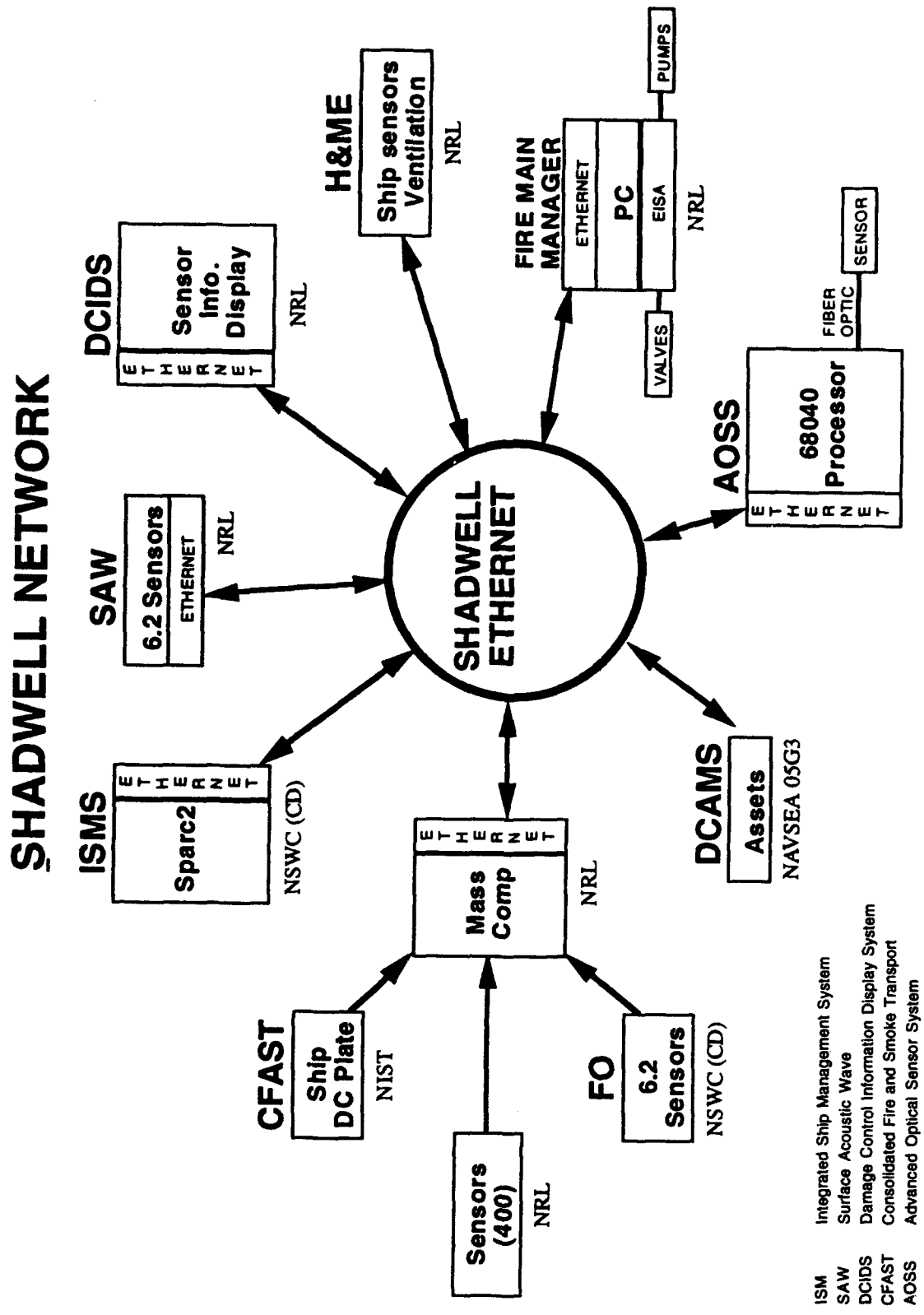

Fig. 1 — Planned Ethernet configuration for ex-USS *Shadwell*

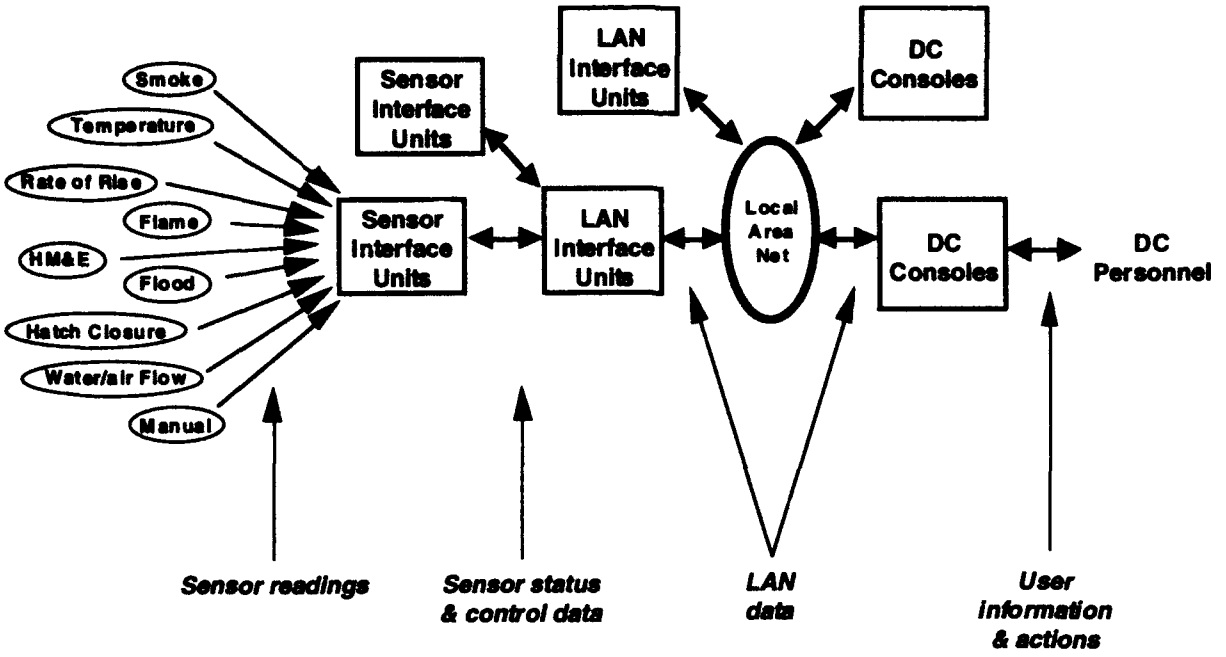

Fig. 2 — Multiple sensors, interface units, and consoles are connected through the local area network in the DCIDS distributed architecture

In this distributed architecture design, it is important that the methods of transferring data between sensors, consoles, and other equipment be designed in a way that meets the specific requirements and constraints of damage control operations. Reference 5 describes the interface protocols and the implementation of these techniques has been included in the *DCIDS program for the Shadwell* described here.

THE DAMAGE CONTROL INFORMATION DISPLAY SYSTEM PROGRAM

The *Shadwell* implementation of the DCIDS program was developed with special consideration for the constraints of using low-cost computer hardware with an operating system that is familiar to many users having only minimal computer experience. It was also imperative that the computer be readily configured for, and connected to, the *Shadwell*'s EtherNet LAN by using off-the-shelf interface boards. A PC-compatible computer running MS-DOS and Windows was selected for its low cost, familiarity, and the availability of various types of interface cards. Using the Windows environment was considered important for compatibility with future systems, and to provide memory management and event-driven tasking capabilities.

System Requirements

A PC-compatible computer was selected as the host for the *Shadwell* implementation of DCIDS. Specific versions of the operating system, window management system, and support programs are required for it to function correctly. The operating system is the MicroSoft Disk Operating System (MS-DOS) Version 5.0. The MicroSoft Windows Version 3.1 window management system is used to take advantage of its advanced windowing techniques and graphical objects and structures. These are standard system components for many PC-compatibles and are usually supplied with the purchase of any new system.

The monitor must be configured for SVGA mode with a screen size of 800 × 600 pixels and 16 colors. Smaller screen configurations will cause an alert message to appear before program execu-

tion begins, to permit the program to be aborted. With an incorrect screen size, the program will run, but much of the information will not be visible, making program control difficult or unpredictable.

Ethernet hardware and software are not generally included as part of a basic computer system. An EtherNet interface card with a ThinNet connector is required to connect to the *Shadwell*'s network. EtherNet cards are typically shipped with diagnostic software for testing the hardware, but additional software, called a packet driver, is also required to use the Transport Control Protocol/ Internet Protocol (TCP/IP) networking functions. This packet driver, which is written for a specific Ethernet card, is usually available from the EtherNet card manufacturer or through public domain software suppliers. It provides the interface required to perform TCP/IP communications. To use DCIDS, the correct packet driver for the specific EtherNet card must be installed.

The software development environment used by DCIDS is Borland's Turbo Pascal for Windows (TPW) Version 1.5. This product is not required to run the DCIDS executable program, but it is necessary if any modifications to the source program are planned. TPW is an object-oriented Pascal with full support for Windows objects and dynamically linked libraries (DLLs). The use of DLLs allows functionality to be added to programs without requiring source code modules for the libraries. Networking software support is provided through a DLL that accesses the packet drivers to perform the network data transfers. The networking package used by DCIDS is the TCP/IP for Windows product from Distinct Corp. This package provides both the libraries and the Pascal header files that are required by DCIDS.

Program Operation

The operation of the DCIDS program is based on the DCIDS prototype described in Ref. 2, but several changes have been made to adjust for system configuration differences. The most obvious difference is the screen size. Because the *Shadwell* program is written for standard SVGA graphics on a PC, the amount of information and resolution of the graphics is reduced. To compensate for the reduced screen size, only a portion of the deck plans are displayed. The displayed area comprises all the compartments forward of Frame 36 from the 02 Level down to the 4th Deck, which constitutes the normal test area for many of the fire tests performed on the *Shadwell*. The positions of some of the display controls have been changed to fit better in the altered screen layout. The reduced screen size also prevented the display of the side view of the ship, resulting in the deck views being selected from a palette of buttons instead of directly from the side view display.

Because of differences between the appearance of graphics objects created by Windows and those used in the original DCIDS prototype, minor changes to some of the buttons and windows are apparent, but their functionality is essentially unchanged. The keyboard was not used as an input device in the original prototype design, but support for menus and their keyboard equivalences are provided in the *Shadwell* version to preserve standard Windows interface features.

Screen Layout

The screen layout can be divided into four functional areas: the menu bar, the deck view area, the selector palettes area, and the sensor information window. The menu bar uses pull-down menus from which program control, deck view selection, sensor type selection, and MassComp network functions are performed. The palettes area is where the palettes for selecting the deck and detector types are displayed; the sensor information window area is where the window containing detailed information for the sensors is displayed. Figure 3 is an example of the screen layout showing the menu bar, the deck view of the 2nd Deck, the selector palettes for the deck and sensor types, and the sensor information window for a typical sensor.

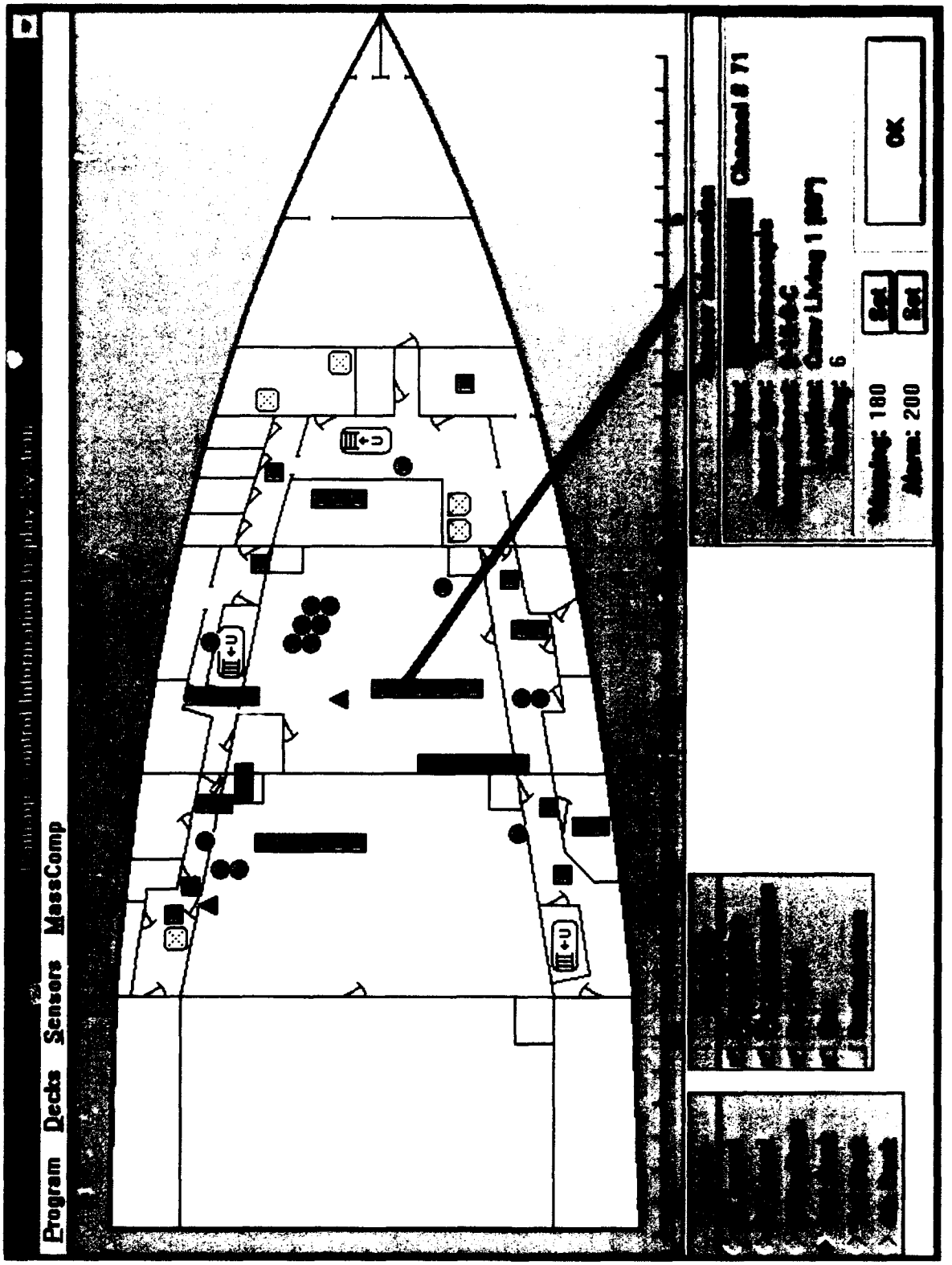

Fig. 3 — The DCIDS screen layout comprises a menu bar, a deck view area, selector palettes for the deck and sensor types, and a sensor information window

The largest portion of the screen is occupied by the deck view display area where the plan view of the selected deck and its corresponding sensors are displayed. The sensor symbols are overlaid on the plan view, with the position of the sensors determined by the data in the sensor configuration data file. The ruler along the bottom of the plan view gives approximate frame numbers for the athwartship bulkheads. The *Shadwell* instrumentation typically uses vertically stacked sensors such as thermocouples to measure temperature at different heights, but this vertical stacking cannot be shown in the plan view, so the sensors are displayed adjacent to each other to allow access to the individual sensor symbols.

Menus

Four primary menus are displayed on the menubar that appears across the top of the screen. These menus and their corresponding menu items are shown in Fig. 4. The first menu, labeled Program, provides program control functions. The About DCIDS ... item displays the version number dialog box. The Audible Alarm item enables and disables the audible alarm function. The beep sound that normally occurs for every warning or alarm signal may become annoying during fire tests, so the ability to disable the audible alarm is provided. A check mark next to the menu item indicates that the audible alarm is active. The Exit item terminates execution of the DCIDS program and returns to the Windows operating system.

Keyboard equivalences for selecting menu items are provided through the standard Windows mechanism. An underlined character in any menu label indicates that the menu may be selected by pressing the Alt key followed by the character that is underlined. A pull-down menu appears, and any of the items in the menu can be selected by typing its corresponding underlined character. For example, the About DCIDS ... item in the Program menu can be activated by pressing the Alt key followed by the P key (to activate the Program menu), followed by the B key (to select the About DCIDS ... item). Menu items without underlined characters can be accessed by pressing the Alt key, followed by the arrow keys to position the menu highlighting over the desired item, then the Enter key to select the item. Function key equivalences are also provided for some menu items; they are indicated by the function key name to the right of the menu item label. When modifier keys such as Ctl or Shift are used in combination with the function key, the label is displayed, for example, as Ctl+F1.

The second menu, labeled Deck, is used to select the deck to be displayed on the screen. The decks from which the user can select are: 02 Level, 01 Level, Main Deck, 2nd Deck, 3rd Deck and 4th Deck. Function keys can be used for selecting a deck, where F1 selects the main deck (i.e., the 1st deck), F2 selects the 2nd deck, F3 selects the 3rd deck, and F4 selects the 4th deck. To select the 01 level, the Ctl+F1 key combination is used, and Ctl+F2 is used to select the 02 level. Only one deck can be viewed at a time, and a check mark next to the menu label indicates which deck is currently displayed.

The Sensors menu selects the types of sensors to be displayed on the deck views. The sensor types are Obscuration meters, Thermocouples, Switch closures, Gas analyzers, and Radiometers. Any combination of sensor types can be viewed at the same time. A check mark next to the menu label for a particular sensor type indicates that that sensor type is currently displayed (if any are present on the selected deck).

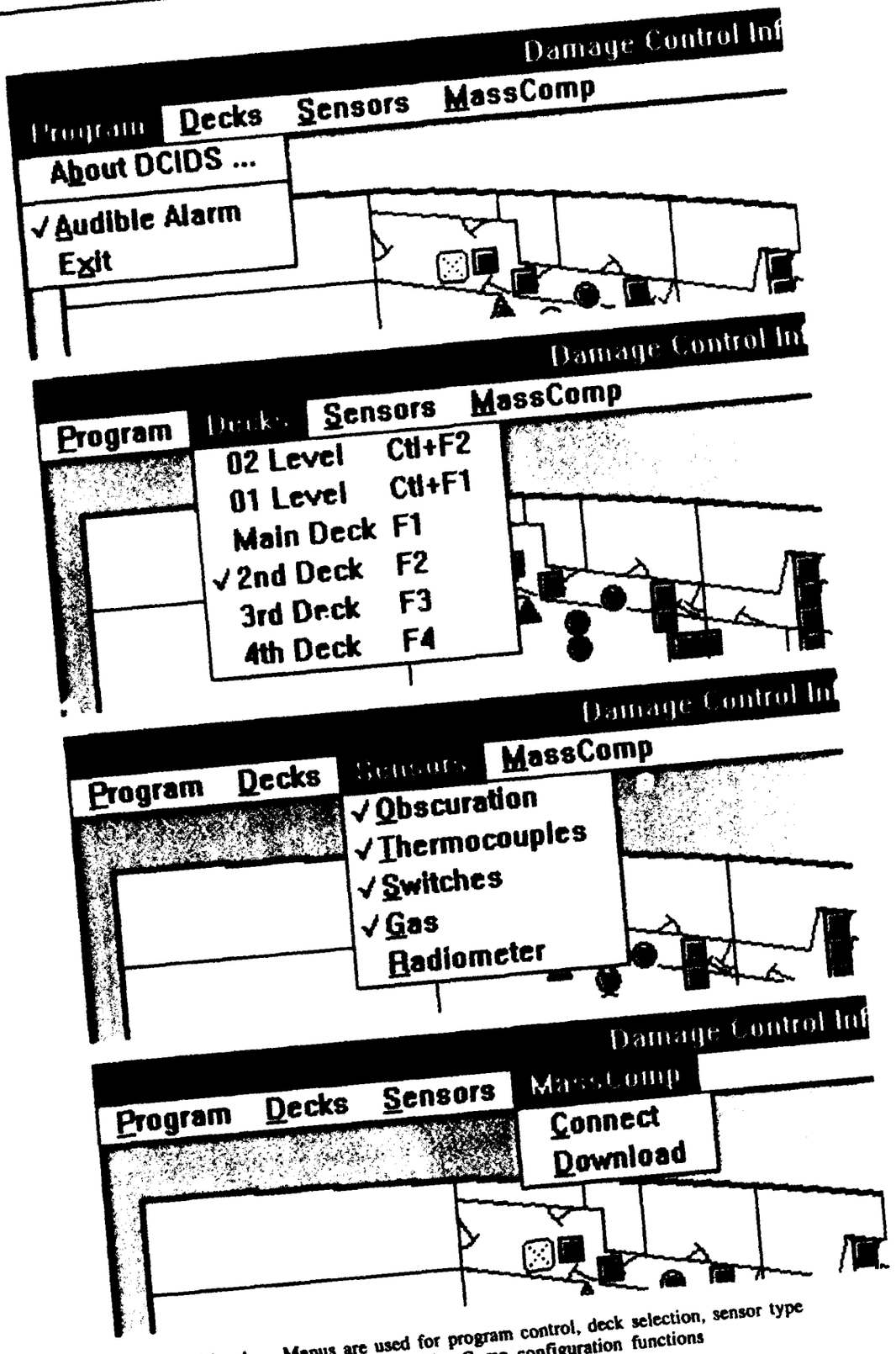

Fig. 4 — Menus are used for program control, deck selection, sensor type selection, and MassComp configuration functions

The **MassComp** menu performs manual initiation of networking and configuration functions for the MassComp computer. When the **Connect** item is selected, the DCIDS program will attempt to establish a connection with the MassComp computer. This step is usually not necessary because the DCIDS program attempts to establish this connection during its initialization process. The **Download** item initializes all the MassComp channel monitoring functions from the settings given in the DCIDS program. This function is also performed during initialization but is provided here to permit manual operation.

Selector Palettes

The Decks and Sensors selector palettes, shown in Fig. 5, provide the same functionality as the Decks and Sensors menus. They are used to provide a quick visual reference of the current display characteristics without the need to pull down a menu. The palettes can be used instead of the menus by clicking on the desired deck or sensor type. The Decks palette uses the button style that is referred to as a radio button, where only one item from the group can be selected at a time. Selecting one deck causes all others to be unselected. The Sensors palette uses the button style called a check box, where any combination of buttons can be selected or unselected. All selected items have a check mark in the box beside the label, and unselected items have only an empty box.

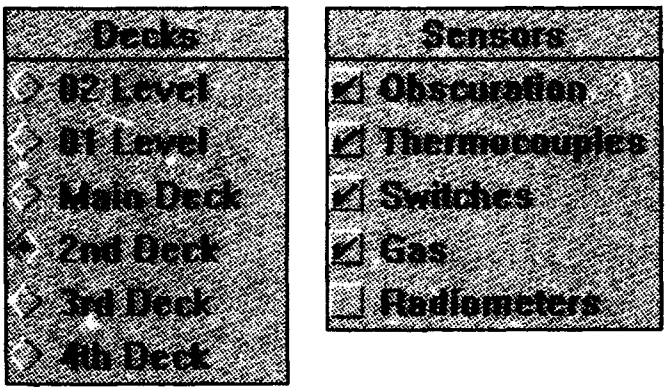

Fig. 5 — Palettes are used for selecting the deck and sensor types to be displayed

Sensor Information Window

To display detailed information about a sensor, the user simply clicks the left mouse button on the sensor symbol. The sensor information window shown in Fig. 6 appears; this provides detailed information about the sensor. The sensor information window is not displayed if a sensor has not been selected. When a sensor is selected, a thick line leading from the sensor to the sensor information window is drawn to serve as a reference pointer. The sensor information includes the sensor status, type, location, reading, MassComp channel number, and the warning and alarm level settings. Only the warning and alarm level settings are configurable by the user. The sensor status is displayed with a color-coded background and can be either Normal (green), Warning (yellow), Alarm (red), Off-line (black), or Maintenance required (blue). The channel number is the I/O channel on the MassComp computer to which this sensor is connected. The sensor location is given by its compartment designation along with the description of the compartment. Additional information may be included in the description such as the height of the sensor or the type of gas being analyzed. The reading is the numerical value of the sensor in standard units for the particular sensor type.

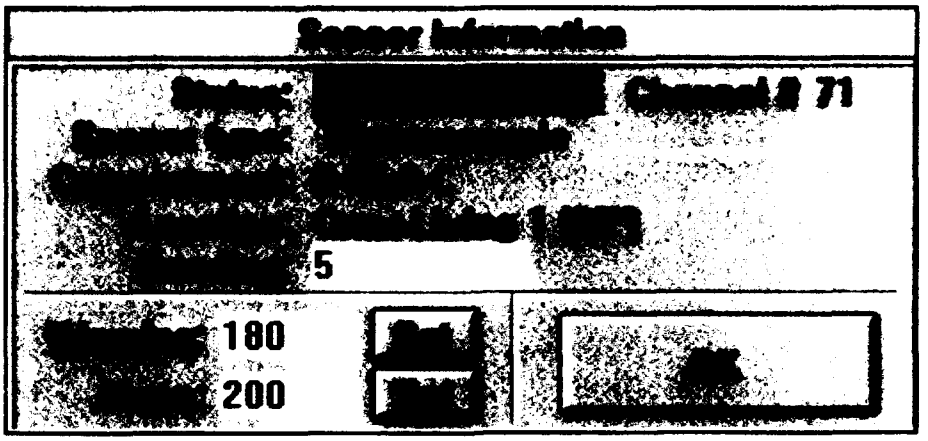

Fig. 6 — The sensor information window provides detailed information about a sensor and allows the warning and alarm level settings to be changed

The warning and alarm level indicators in the sensor information window display the current settings and allow the user to change the settings. New values can be typed into the boxes and sent to the MassComp by pressing the adjacent Set button. The OK button is used to close the window and erase the line leading to the sensor. If the warning or alarm level settings are changed without the corresponding Set button being pressed, the user is prompted to change the setting or cancel the operation with an alert box such as is shown in Fig. 7.

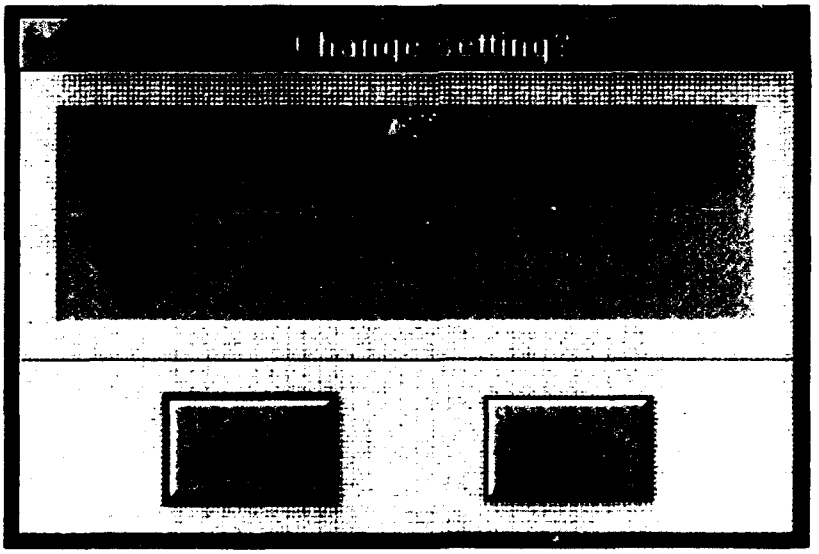

Fig. 7 — An alert box prompts the user for a response if the warning or alarm level settings are changed without sending the changes to the MassComp computer

Sensor Configuration File

All of the sensor information used by the DCIDS program is retrieved from a sensor configuration file that can be changed to reflect different test conditions. This file is a text file that can be edited with standard text editors such as the Windows Notepad program. The configuration file is named DCIDS.DAT and must reside in the same directory as the DCIDS program. Figure 8 shows an example portion of a sensor configuration file. Each line in the file describes a single sensor, or provides a comment indicated by an exclamation point in the first character of the line. The description of each sensor is composed of the MassComp channel number, the sensor type, the warning level

```
! DCIDS.DAT
! channelNumber,type,warningLevel,alarmLevel,compartment,descriptor,x,y
!
159, Thermocouple, 180, 200, 2-15-2-L, Passage (60°), 351, 73
158, Thermocouple, 180, 200, 2-15-2-L, Passage (18°), 351, 85
141, Thermocouple, 180, 200, 2-15-2-L, Passage, 351, 97
142, Thermocouple, 180, 200, 2-15-2-L, Passage, 351, 109
144, Thermocouple, 180, 200, 2-15-2-L, Passage (60°), 436, 115
!
59, Obscuration, 55, 60, 2-15-2-L, Passage (120°), 386, 83
!
72, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (OH), 356, 193
71, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (98°), 356, 205
70, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (70°), 356, 217
69, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (60°), 356, 229
68, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (36°), 356, 241
67, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (18°), 356, 253
!
127, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (108°), 308, 223
126, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (90°), 308, 235
125, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (72°), 308, 247
124, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (54°), 308, 259
123, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (36°), 308, 271
122, Thermocouple, 180, 200, 2-15-0-C, Crew Living 1 (18°), 308, 283
!
111, Obscuration, 55, 60, 2-15-0-C, Crew Living 1 (60°), 421, 233
!
175, Radiometer, 65, 70, 2-15-0-C, Crew Living 1, 320, 188
!
169, Switch, 0, 1, 2-15-0-C, Crew Living 1 (hatch), 349, 166
!
184, Gas, 20, 50, 2-15-0-C, Crew Living 1 (CO @ 100°), 385, 136
183, Gas, 20, 50, 2-15-0-C, Crew Living 1 (CO @ 24°), 385, 148
188, Gas, 35, 40, 2-15-0-C, Crew Living 1 (CO2 @ 100°), 397, 142
187, Gas, 35, 40, 2-15-0-C, Crew Living 1 (CO2 @ 24°), 397, 154
```

Fig. 8 — Example portion of a sensor configuration file

setting, the alarm level setting, the compartment in which it is located, the description, and the x and y coordinates on the plan view display. Each of the sensor parameters are separated by a comma; leading spaces are ignored but imbedded spaces are not.

The MassComp channel number is used by the DCIDS program to query the MassComp computer about the sensor. It is the user's responsibility to ensure that the channel number in the configuration file is correct for each sensor. Incorrect sensor types or locations cannot be checked by the DCIDS program, and may produce unexpected program performance.

Sensor types can be only obscuration, thermocouple, switch, gas, or radiometer, corresponding to the sensor types shown in the sensor type selector palette. Other sensor types may be added in a future release of DCIDS, but currently, use of any other sensor type will produce an error.

The warning and alarm level settings are specified in the units used by the MassComp for reporting on the particular sensor type. No conversion is performed by the DCIDS program. To force the MassComp monitoring function to report an alarm condition with no intervening warning, the warning and alarm level settings should be set to the same value. Alarm conditions take precedence over warning conditions; if an alarm condition occurs at the same time as a warning condition, only the alarm is reported.

The compartment number must be given in the standard deck and frame number nomenclature used for shipboard compartments. It is important to note the difference between the nomenclature

used for decks above the main deck versus decks below it. Compartments in the upper decks always begins with a leading zero, as in 02-15-2-L (a compartment on the 02 Level), whereas the lower deck compartments never have the leading zero, as in 2-15-2-L (a compartment on the 2nd Deck). The DCIDS program checks for the leading zero in the compartment number to determine which deck the sensor is displayed on. The actual location of the compartment on the deck is not checked since fine adjustment of the sensor position is determined by sensor's x and y coordinates.

The sensor description can be any text. It typically is a description of the compartment or specific information about the sensor that uniquely identifies it. A useful description may include the height of the sensor or the type of gas being analyzed.

The x and y coordinates of the sensor are used for the actual location of the sensor symbol on the plan view display. The method for determining the coordinates for a sensor is to click the right mouse button on the plan view display at the desired location, causing the x and y coordinates of that location to be displayed in the upper left corner. These numbers are used as the coordinates in the configuration file. When placing sensors in close proximity to each other, it is useful to note that the sensor symbols are all 12 pixels in width and height. To create a "stack" of sensors, only the position of the first sensor needs to be determined since an offset of 12 can then be added to the position of the other sensors to position them correctly.

Sensor configuration files can be customized to monitor only those sensors of interest for a particular test, or only the sensors in a particular area of the ship. The DCIDS program can be run on multiple computers connected to the *Shadwell* EtherNet. Different views of the evolution of the fire test are then provided through the use of customized configuration files. This technique can be useful in situations where the scientist may be interested in a large amount of highly instrumented sensor information for research purposes whereas a damage control trainee may only have access to limited sensor information that is representative of an actual shipboard configuration.

Plan View Bitmap Files

The DCIDS program uses six bitmap files for displaying the plan views of the various decks. These files must be located in a subdirectory named DECKS that resides in the same directory as the DCIDS program. These files are standard Windows .BMP files but must not be modified in any way by the user.

COMMUNICATION WITH THE MASSCOMP COMPUTER

The DCIDS program provides a user-friendly operator's console for monitoring and controlling sensor parameters, but it does not communicate directly with any sensors. The MassComp computer on the *Shadwell* is the primary data acquisition device and provides the interface to the sensors. The DCIDS program accesses the sensor information through a communication protocol designed for damage control systems that has been tailored for use with the MassComp.

Communication Protocol

The DCIDS program communicates with the MassComp by using a protocol that follows the guidelines set forth in Ref. 5 to assure timely reporting of sensor information. Reference 4 describes the protocol used between the MassComp and the DCIDS program; it is included as an appendix to this report. The protocol is based on the use of TCP/IP connection-oriented stream sockets that provide guaranteed delivery of data between connected devices. Connection-oriented communication provides the network integrity and guaranteed delivery checks that are required by damage control systems.

The MassComp runs a background process that listens for connection requests on the network port reserved for DCIDS communications. During initialization, the DCIDS program attempts to open a connection with the MassComp on this port. If the initial connection cannot be made, the alert box shown in Fig. 9 is displayed on the screen, but the program continues to run. This situation may occur if the DCIDS program is started before the MassComp computer is turned on. The DCIDS program will continue to attempt to open a connection until a connection is established. The alert box is displayed only for the initial connection attempt. Later connection attempts are performed without any user notification. The user can also initiate a connection manually from the MassComp menu, if desired.

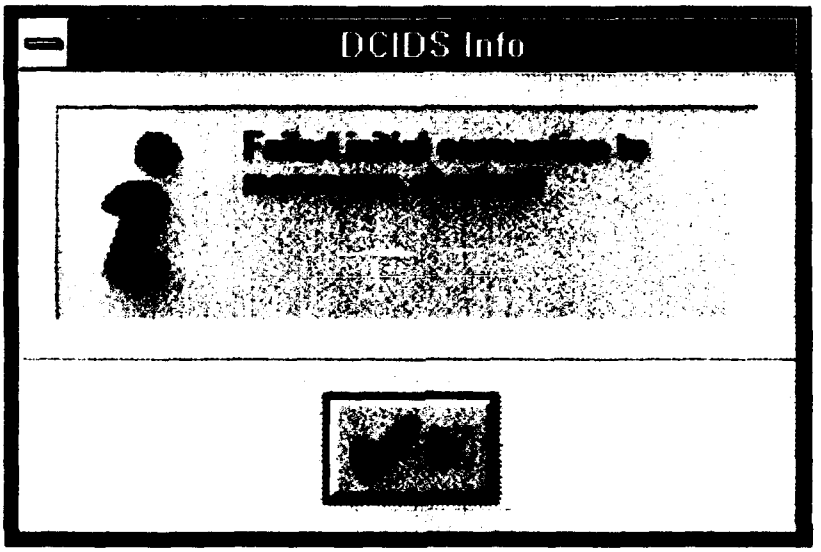

Fig. 9 — An alert box is displayed if the initial connection to the MassComp computer fails

Downloading Warning and Alarm Level Settings

After opening a connection, the DCIDS program downloads the warning and alarm level settings and requests readings for all the sensors listed in the sensor configuration file. Since the communications protocol supports multiple connections, and the DCIDS program may be running on more than one computer, the DCIDS program requests the current warning and alarm level setting before sending any new settings. This permits a second DCIDS program to begin execution after one has already been running, and the settings established by the first program will be preserved. The second program will download only settings that have not already been established. The download process can also be initiated manually from the MassComp menu, if desired.

Sensor Polling

Once the warning and alarm level settings have been downloaded to the MassComp, no further monitoring of sensors is required. However, the DCIDS program performs round-robin polling of the sensors to keep its internal data up to date. The round-robin schedule simply polls the sensors at one second intervals in the sequence given in the configuration file. Round-robin polling is suspended when the sensor information window is displayed, but polling resumes when the window is closed.

After clicking on a sensor symbol, the DCIDS program displays the last reported sensor reading, warning level, and alarm level contained in its internal data, and requests the current values from the MassComp computer. The new values are displayed in the sensor information window as soon as they are received from the MassComp. The communication between the MassComp and DCIDS is very fast, and the new values are displayed as soon as the sensor information window is drawn. While the sensor information window is visible, the round-robin polling is suspended, and the sensor being displayed is continuously polled. This permits the user to observe changes to the sensor reading. The continuous polling of the sensor has no effect on the warning and alarm reporting functions, and the plan view display is updated to reflect changes in the status of any sensors.

FUTURE CONSIDERATIONS

This version of the DCIDS program provides sensor monitoring and reporting capabilities but does not include control of the fire mains system as was demonstrated in the prototype described in Ref. 2. Although the MassComp has the ability to control a limited number of output devices, this capability has not been implemented on the *Shadwell*. When control of devices such as valves, pumps, fans, and dampers is implemented on the MassComp, the ability to control these devices can be easily integrated into the DCIDS program.

To better demonstrate the DCIDS concept of a distributed architecture of consoles and sensor interfaces, it would be useful to integrate other sensor interface devices into the *Shadwell* network. The current configuration uses a network comprising only two nodes, which is a minimal network configuration. Integration of systems such as the Fire Main Manager, being developed for an EISA computer, into the Shadwell network will provide an opportunity to further demonstrate the distributed architecture concept and expand the capabilities of the DCIDS program.

CONCLUSIONS

The Damage Control Information System is the product of an exploratory development project that has developed the use of graphical user interfaces and network interface protocols for distributed architecture damage control systems. The implementation of DCIDS for the *Shadwell* has provided an opportunity to begin the transition of these ideas from the concept development stage to actual implementation. DCIDS for the *Shadwell* is a prototype system that effectively demonstrates the feasibility and utility of these concepts. It is proof that these techniques will provide a fast and effective way of managing the Navy's complex damage control systems of the future.

ACKNOWLEDGMENTS

The author thanks Dr. Patricia Tatem and Dr. Frederick Williams of the Navy Technology Center for Safety and Survivability (NRL Code 6180) for their support of this effort. This report is submitted in partial fulfillment of Milestone 2, Task 5 of the Damage Control Project (RH21S22) of the Surface Ship Technology Block Program (ND1A/PE62121N). The work described is sponsored by the Office of Naval Research (ONR46) and performed by the Naval Research Laboratory (NRL Code 5535).

REFERENCES

1. H. K. Whitesel, W. F. Zeller, and C. A. Miller, "Sensor Requirements and Technology for Damage Control," David Taylor Research Center Report DTRC/PAS-90-7, June 1990.

2. D. L. Tate, "A Graphical User Interface Design for Shipboard Damage Control," NRL Report 9355, Aug. 26, 1991.

3. D. L. Tate and F. W. Williams, "Ethernet Options for the Ex-USS Shadwell," NRL Letter Report 6180/393A.1, July 1, 1993.

4. D. L. Tate, T. Riddle, and F. W. Williams, "Real Time Data Access aboard the Ex-USS Shadwell," NRL Letter Report 6180/54, Feb. 8, 1993.

5. D. L. Tate, "Interface Protocol Requirements for Shipboard Damage Control Systems," NRL Report NRL/FR/5533—92-9525, Sept. 30, 1992.

Appendix

REAL TIME DATA ACCESS VIA ETHERNET ABOARD THE EX-USS *SHADWELL*

David L. Tate
Naval Research Laboratory, Code 5535

Tom Riddle
Hughes Associates, Inc.

F. W. Williams
Naval Research Laboratory, Code 6183

INTRODUCTION

The ex-USS *Shadwell* is the U.S. Navy's full-scale fire research and test ship from which damage control and ship survivability investigations, analyses, and evaluations are performed. The ex-USS *Shadwell* is the decommissioned USS *Shadwell* (LSD-15) landing ship dock and is berthed at the U.S. Coast Guard's Fire and Safety Test Detachment, Little Sand Island, Mobile, Alabama. The coordinator and custodian of the ex-USS *Shadwell* is the Navy Technology Center for Safety and Survivability at the Naval Research Laboratory (NRL), Code 6180.

The ex-USS *Shadwell* is highly instrumented and controlled for acquisition, processing, and storage of data, both in real time and for later processing. Up to 400 sensors, which measure temperature, obscuration, heat, radiation, gaseous products, air and liquid velocity and pressure, and switch closure state can be connected to the MassComp computer that serves as the central data acquisition and collection system. Sensor data are acquired in real time during full-scale fire tests and saved to the MassComp disk for later retrieval. A new capability for real-time data collection has been added to the MassComp data acquisition software that uses an Ethernet Local Area Network (LAN) to transfer data to other computers. This new software allows other computers to request data from the MassComp across the Ethernet during the full scale fire tests.

The Ethernet software also provides networking capabilities such as file sharing, rapid data transfer, and remote login. Standard networking utilities such as FTP (File Transfer Protocol) and Telnet (remote login) are supported on the MassComp computer and the Packard Bell PC located in the Shadwell's control room. The PC has a Windows version of the FTP and Telnet programs to allow transfer of files to and from the MassComp onto PC-compatible floppy disks.

ETHERNET CONFIGURATION

The Shadwell Ethernet uses RG-58 coaxial cable, commonly referred to as ThinNet, which uses standard BNC connectors. Any cables, connectors, or adapters necessary to connect equipment to the Ethernet are the user's responsibility and are not available from ship's inventory. Installation and/or removal of any equipment on the Ethernet must be approved by ship's personnel to insure smooth operation of the network. Nodes will be assigned by NRL. Installation or removal of equipment during fire tests is strictly forbidden.

Connecting to the MassComp via TCP/IP

The MassComp runs a background process (called a daemon) that listens for TCP/IP connection requests on the Ethernet. The Transmission Control Protocol (TCP) is a connection-oriented protocol that uses stream sockets at both the server and client ends of the network connection. The MassComp daemon functions as a server, and any computer trying to access it is a client. The port number for access to the daemon is 5030. The Internet Protocol (IP) address for the MassComp is 89.0.0.1, and its host name is "masscomp.shadwell" (or simply "masscomp"). The Packard Bell PC in the control room has an IP address of 89.0.0.2 with a host name of "controlroom.shadwell" (or simply "controlroom"). Several other IP addresses have been assigned for temporary use by other computers. Before connecting to the Ethernet on the Shadwell, all computers must be assigned IP addresses by ship's personnel.

Client computers may open and close connections to the MassComp before, during, and after live fire tests. Data retrieved from the MassComp reflects only current conditions. No data history is saved in memory since this information can be retrieved from the fire test data recorded on the MassComp's disk.

Using the Data Retrieval Protocol

The data retrieval protocol is used to transfer information between the MassComp computer and any other computers connected on the Ethernet. The protocol allows users to retrieve readings, initiate and cancel alarm detections, and set alarm trip points for a user-selectable set of sensors.

Each message passed between the MassComp and any client computer is based on a sixteen-bit word size for all its commands, channel numbers, and a 32-bit floating point number for its data values. The first word of each message is called the opcode, and is followed by a variable number of parameter words. The length and content of the message that accompanies each opcode is determined by the opcode. Table 1 is a list of the valid opcodes and their parameters for messages sent from a client computer to the MassComp. Table 2 is a list of the valid opcodes and their parameters for messages sent from the MassComp to a client computer. Note that the messages have different meanings depending on whether they originate at the client or the MassComp. A detailed description of each message is provided in the following section to clarify their meaning. The opcode contained in a message determines the message type (i.e., *SetValue* messages are messages that contain the *SetValue* opcode). Messages that originate at the client are referred to as client messages, and message that originate at the MassComp are MassComp messages.

It is important to note that all messages (except *NOOP*) refer to a channel number. This number is the corresponding MassComp channel number, and inclusion of this number in all messages allows both requested data values and unrequested alarm indications to be intermixed in the message stream without confusion. A user-selected set of channel numbers can be used to permit client computers to collect data only for those sensors of interest during a particular test.

Client Messages

Client messages are used to request sensor readings, and control the warning and alarm detection software in the MassComp. Sensor readings are provided as a floating point value in appropriate units for the particular sensor. Warning and alarm detection provides two levels of sensor alert conditions. Warning detection provides the lowest alert indication, and is typically associated with the color yellow. Alarm detection is a higher alert level that is typically associated with the color red. Sensor readings that have not reached the warning or alarm levels are considered in their normal operating range and are associated with the color green. Sensors that have not been assigned a warning or alarm setting have an undefined status and may be associated with the color black (i.e., no color).

Table 1. CLIENT MESSAGES

Opcode	Parameters	Description
NOOP=0	none	No operation - used for connectivity check
SetValue=1	channel, value	Command to set the value for this channel
GetValue=2	channel	Request for the value and status for this channel
SetWarningLevel=3	channel, value	Command to set the warning level for this channel
GetWarningLevel=4	channel	Request for the warning level for this channel
DisableWarning=5	channel	Command to disable warning detection for this channel
SetAlarmLevel=6	channel, value	Command to set the alarm level for this channel
GetAlarmLevel=7	channel	Request for the alarm level for this channel
DisableAlarm=8	channel	Command to disable alarm detection for this channel

Table 2. MASSCOMP MESSAGES

Opcode	Parameters	Description
NOOP=0	none	No operation - used for connectivity check
SetValue=1	channel	Used only to NACK a SetValue command (see Error Conditions section)
GetValue=2	channel, value, status	Report of the value and status for this channel
SetWarningLevel=3	channel	Used only to NACK a SetWarningLevel command (see Error Conditions section)
GetWarningLevel=4	channel, value	Report of the warning level for this channel
DisableWarning=5	channel	Used only to NACK a DisableWarning command (see Error Conditions section)
SetAlarmLevel=6	channel	Used only to NACK a SetAlarmLevel command (see Error Conditions section)
GetAlarmLevel=7	channel, value	Report of the alarm level for this channel
DisableAlarm=8	channel	Used only to NACK a DisableAlarm command (see Error Conditions section)
NotTripped=9	channel	Report indicating neither the warning nor the alarm detectors are tripped
WarningTripped=10	channel	Report indicating the warning detector is tripped but the alarm indicator is not tripped
AlarmTripped=11	channel	Report indicating that the alarm indicator is tripped

To request the reading for a particular sensor, a client sends a message to the MassComp composed of the *GetValue* opcode followed by the appropriate channel number. The reply from the MassComp would be composed of the *GetValue* opcode, followed by the channel number, the value (reading) of the sensor, and the appropriate status. Status values are *Undefined*=0, *Normal*=1, *Warning*=2, and *Alarm*=3. All sensors have an undefined status if no warning or alarm levels have been assigned to them.

The *SetValue* command is provided to allow client computers to issue commands to open/close valves, start/stop pumps, or set equipment parameters. This command is used for future capabilities as the MassComp has the capability of controlling up to 48 devices.

The warning detection software in the MassComp is controlled by the *SetWarningLevel* and *DisableWarning* commands. To set a warning level for a particular channel, the client sends a message composed of the *SetWarningLevel* opcode, followed by the channel number and the value of the sensor reading above which a warning condition should be declared. The MassComp will monitor the sensor and declare a warning condition by sending a *WarningTripped* message to the client (see MassComp Messages section) when the warning level has been exceeded. To retrieve the current warning level from the MassComp, the client issues a message composed of the *GetWarningLevel* opcode followed by the channel number of interest. To discontinue the warning detection software in the MassComp for a particular channel, the *DisableWarning* command is issued with the appropriate channel number.

The *SetAlarmLevel*, *GetAlarmLevel*, and *DisableAlarm* messages control alarm detection in the MassComp in the same manner as the corresponding *SetWarningLevel*, *GetWarningLevel*, and *DisableWarning* commands control warning detection.

The *NOOP* command performs no operation but is included to provide a network connectivity check. The *NOOP* message is a single word message with no associated channel number. Although the TCP/IP protocol provides connectivity monitoring and guaranteed delivery of packets, systems with only a subset of the TCP/IP protocol may use the *NOOP* message as a means of manually verifying connectivity. No response is generated by the recipient of the *NOOP* message, but the acknowledged receipt of the packet provided by TCP/IP can be used as a connectivity check. All client and server devices must be able to receive a *NOOP* message.

MassComp Messages

MassComp messages are used to reply to requests for sensor readings, warning levels, and alarm levels, as well as to report alarm trip conditions to client computers. Under normal conditions, some opcodes are never sent from the MassComp to the client computers, but clients must be able to receive all opcodes to insure processing of error conditions.

The MassComp's *GetValue* message is the reply to a client's *GetValue* message requesting a sensor reading. The parameters included in this message are the channel number, the sensor value (reading), and the status code. The value is in units appropriate for the particular sensor and the status code is *Undefined*, *Normal*, *Warning*, or *Alarm*, as described above.

The MassComp's *GetWarningLevel* message is the reply to a client's *GetWarningLevel* request. This message allows clients to read the current settings of the MassComp's warning levels table. This is useful when clients connect to the MassComp after the warning level tables have already been initialized.

The MassComp's *GetAlarmLevel* message is the reply to a client's *GetAlarmLevel* request. This message allows clients to read the current settings of the MassComp's alarm levels table. This is useful when clients connect to the MassComp after alarm level tables have already been initialized.

The *NotTripped* message is sent by the MassComp to indicate that a warning condition (or alarm condition if no warning level has been set) has transitioned from an alert condition back to a normal condition. This message is sent only once when the transition occurs.

The *WarningTripped* message indicates that the Warning level has been exceeded for the channel and the sensor is now in a warning condition. This message is sent only once when the transition occurs.

The *AlarmTripped* message indicates that the Alarm level has been exceeded for the channel and the sensor is now in an alarm condition. This message is sent only once when the transition occurs.

Error Conditions

Under normal conditions, no explicit acknowledgment of the receipt of a valid message is provided. The TCP/IP protocol provides guaranteed delivery of messages, and delivery failure is handled through its own error reporting mechanisms. However, incorrect use of messages between a client and the MassComp (or other computers) cannot be detected in the TCP/IP software. Invalid messages may be either invalid because the function is not supported or the corresponding channel number is invalid. The data retrieval protocol provides an error reporting mechanism through the use of a non-acknowledgment (called a *NACK*) of a message in error. A *NACK* is a message that is sent to the originator of the invalid message. It is composed of an opcode followed by a channel number. The opcode is created by negating the opcode of the message in error (e.g., a *NACK* of a *SetValue* message uses an opcode value of -1). If the channel number is invalid, both the opcode and the channel number in the *NACK* message are negated.

*NACK*s are used to indicate that an invalid request or command has been issued. Sending a *SetValue* command to the MassComp is an example of an invalid command because the MassComp has no settable devices. The *SetValue* command may be a valid command for other (future) device servers, such as a fire pump controller. A *GetValue* command for channel #500 would produce a *NACK* with both the opcode and channel number negated because the MassComp only has 400 channels. Note that some of the opcodes listed in Table 2 are only transmitted by the MassComp when an error occurs. Client programs should allow for reception of all opcodes (both positive and negative) that are listed in Table 2.

Typical Setup and Operation

The first steps taken by a client computer to establish communication with the MassComp are to create a network socket and open a connection. Using the Berkeley Software Distribution (BSD) UNIX conventions, the socket is created by calling `socket()` as in:

```
sd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP); /* create the socket */
```

where `AF_INET` is the InterNet address family, `SOCK_STREAM` is the stream socket type, and `IPPROTO_TCP` is the TCP protocol. The returned value that is assigned to `sd` is the socket descriptor. Note that some systems use non-blocking calls to socket functions. This means that the call returns immediately, and another function is called to check the input/output (I/O) completion status.

The examples given here assume all function calls are blocking, so that the return from the function is blocked until all I/O is completed.

A network address structure of type `sockaddr_in` is associated with the socket connection to the MassComp as follows:

```
struct sockaddr_in srv_addr;  /* srv_addr is the net addr struct for server */
srv_addr.sin_family = AF_INET;                  /* address family for server */
srv_addr.sin_addr.s_addr = inet_addr("89.0.0.1");/* IP address for MassComp */
srv_addr.sin_port = htons((unsigned short) 5030);     /* port number to use */
```

The client then connects to the MassComp via the `connect()` call as in:

```
err = connect(sd, &srv_addr, sizeof(srv_addr)); /* open MassComp connection */
```

Once connected, the MassComp and client can exchange messages back and forth with either standard `read()` and `write()` calls, or socket function calls `recv()` and `send()`.

If a client program only wants to take periodic readings of a set of sensors, a simple program can be written that transmits *GetValue* requests for the channels of interest, and processes incoming *GetValue* replies from the MassComp. Clients can control the sampling rate of the data by delaying the *GetValue* requests. The fastest scan rate available from the MassComp is once per second.

```
#define CMD_GETVALUE 2
short channel[NUM_CHANNELS];  /* declaration of array for channel numbers */
short transmit_buf[1000];   /* socket transmit buffer (arbitrarily large) */
short receive_buf[1000];     /* socket receive buffer (arbitrarily large) */
...       /* initialize channel numbers, set up program timing loop, etc. */
/* beginning of send/receive loops */
for (i = 0; i < NUM_CHANNELS; i++)                /* loop for each channel */
  {
  transmit_buf[2 * i] = CMD_GETVALUE;        /* put GetValue command into
                                                          transmit buffer */
  transmit_buf[(2 * i) + 1] = channel[i];    /* put channel number after
                                                        GetValue command */
  }
num_written = send(sd, transmit_buf, 4 * NUM_CHANNELS, 0);
if num_written != (4 * NUM_CHANNELS)
  printf("Send() failed with error %d.\n",errno);
...
for (i = 0; i < NUM_CHANNELS; i++)                /* loop for each channel */
  {
  num_read = recv(sd, receive_buf, 8, 0);/*get 8 bytes (2 shorts, 1 float)*/
  if num_read != 8 then                          /* should receive 8 bytes   */
    printf("Recv() failed after receiving %d bytes.\n", num_read);
  if receive_buf[0] == CMD_GETVALUE      /* should receive GetValue opcode */
    printf("Channel %d has value = %f and status = %d.\n",
                receive_buf[0], (float) receive_buf[1], receive_buf[3]);
  else  /* problem with opcode */
    /* real programs should have more extensive error checking than this */
    printf("Received bad opcode = %d.\n",receive_buf[0]);
  }
/* end of send/receive loops */
... /* perform any other processing, end of timing loop, etc. */
```

The same technique of filling the transmit buffer and sending the data is performed to set warning and alarm levels. With warnings or alarms enabled, a more elaborate receive function must be

written to be able to accept various opcodes and their corresponding parameters that may be received at arbitrary times. The following example shows a simple (though not very efficient) way of handling various opcodes and message lengths:

```
#define CMD_NOOP 0
#define CMD_GETVALUE 2
#define CMD_WARNINGTRIPPED 10
short opcode, channel_num, status;
float value;
...
/* beginning of arbitrary opcode processing */
num_read = recv(sd, opcode, 2, 0);      /* get opcode */
if num_read != 2 then printf("Recv() failed getting opcode.\n");
switch (opcode) /* use different processing for different opcodes */
  {
  case CMD_NOOP: /* NOOP has no channel number or parameters */
    printf("Received a NOOP.\n"); /* nothing more to do */
    break;
  case CMD_GETVALUE: /* GetValue has channel number and two parameters */
    num_read = recv(sd, channel_num, 2, 0);       /* get channel number */
    if num_read != 2 then printf("Recv() failed getting channel number.\n");
    num_read = recv(sd, value, 4, 0);             /* get channel value */
    if num_read != 4 then printf("Recv() failed getting value.\n");
    num_read = recv(sd, status, 2, 0);            /* get channel status */
    if num_read != 2 then printf("Recv() failed getting status.\n");
    printf("Channel %d has value = %f and status = %d.\n",
                channel_num, value, status);
    break;
  case CMD_WARNINGTRIPPED: /* WarningTripped has channel number only */
    num_read = recv(sd, channel_num, 2, 0);     /* get channel number */
    if num_read != 2 then printf("Recv() failed getting channel number.\n");
    printf("Warning on channel %d.\n",channel_num);
    break;
  ... /* other opcode processing */
  }
...
/* go to beginning of opcode processing
   after completely processing any message */
```

The examples given above provide a basis for an operational program. Obviously, much more extensive code must be written to correctly handle all opcodes, error conditions, opening and closing connections, and any other client functions that may be necessary to accomplish the desired task.

Future Capabilities

This document describes the initial version of the protocol used to retrieve fire test data from the MassComp. Additional capabilities such as block transfers of sensor data, control of output devices, and interclient communication are expected to be supported in future versions.